Principal component analysis

The toy dataset used to illustrate the principal component analysis is about French towns.

The statistical unit is a town. 31 cities have been observed for 34 variables

```
NO2             :    Nitrogen dioxide
DENSITY         :    People per km2
RAINFALL        :    Average annual rainfall (mm)
"MONTH"+r       :    Average monthly rainfall (12 variables)
DAY_RAINFALL    :    Average annual number of rainy days
"MONTH"=dr      :    Average monthly number of rainy days (12 variables)
TEMP            :    Average annual temperature (d° Celcius)
TEMP_RANGE      :    Temperature variation
SUNSHINE        :    Average of sunny days (hours per day)
LATITUDE        :    Latitude
LONGITUDE       :    Longitude
GEO             :    Geographic positions (South=1 / East=2 / West=3 / North=4)
```

In [ ]:
```
Dataset=read.table("FrenchCities.csv", header=T,sep=';',row.names=1)
Dataset
```

Dataframe

The result of read.table is a *dataframe* with attributes.

In [ ]:
```
attributes(Dataset)
```

- Select Variable :

In [ ]:
```
# with the attribute $var
Dataset$NO2

# with the column number
Dataset[,1]

# with the column name
Dataset[,"NO2"]
```

- Select observations :

In [ ]:
```
# Select Rows 10, 11 and 12
Dataset[10:12,]
# Create a subset of Dataset according to the logical expression
subset(Dataset,NO2>50)
# Remove the first five rows
Dataset[-(1:5),]
```

Export a dataframe

The function *write.table* prints the dataframe x to a file

```
write.table(x,file='name.file',sep=';',row.names=T,col.names=T,....)
```

The variables

Variable types

Statistic methods depends if the variable is categorial, type *factor*, or quantitative, type *num* or *int*. *str* gives the variable types.

In [ ]:
```
str(Dataset)
```

The type of variable GEO is integer. We have to transform it into factor with labels :

```
South=1
East=2
West=3
North=4
```

as.xx(var) transforms var into type xx. *is.xx(var)* returns TRUE if var is type xx. *levels(var)* gives the labels of a categorial variable.

In [ ]:
```
# Transform variable GEO into factor
Dataset$GEO=as.factor(Dataset$GEO)
is.factor(Dataset$GEO)
# Display its labels
levels(Dataset$GEO)
```

In [ ]:
```
# Change its labels with the vector c('South','East','West','North')
levels(Dataset$GEO)=c('South','East','West','North')
levels(Dataset$GEO)
```

summary provides numerical indicators (mean, median,...) if the variable is quantitative and the label frequencies for categorial variables.

In [ ]:
```
summary(Dataset)
```

Quantitative variable

- Numerical summarization :

In [ ]:
```
mean(Dataset$NO2) # Mean value
sd(Dataset$NO2)  # Standard deviation
median(Dataset$NO2) # middle value when the values are sorted
quantile(Dataset$NO2)# Split the ordered values into 4 equal parts and gives the points between the quarter
s
cor(Dataset[,1:5]) # correlation matrix of the five first variables
```

- Graphical summarization :

In [ ]:
```
boxplot(Dataset$TEMP,main='Boxplot of TEMP',ylab='TEMP',col='grey')
hist(Dataset$TEMP,main='Histogram of TEMP',xlab='TEMP',col='blue')
plot(Dataset$LATITUDE,Dataset$TEMP,main='TEMP vs LATITUDE',xlab='LATITUDE',ylab='TEMP',col="red",lwd=2) # S
catterplot
text(Dataset$LATITUDE,Dataset$TEMP,row.names(Dataset)) # add the names of the points
pairs(Dataset[,1:5]) # Scatterplot matrix for the five first variables
```

In [ ]:
```
boxplot(Dataset$NO2)
```

Error is not a French city but an outlier. We remove it from the dataset.

In [ ]:
```
Dataset0=Dataset # Dataset0 is the dataset with error

Dataset=subset(Dataset,TEMP>0) # Dataset is the data without Error (since TEMP=0 for Error)

boxplot(Dataset$TEMP,main='Boxplot of TEMP',ylab='TEMP',col='grey')
hist(Dataset$TEMP,main='Histogram of TEMP',xlab='TEMP',col='blue')
plot(Dataset$LATITUDE,Dataset$TEMP,main='TEMP vs LATITUDE',xlab='LATITUDE',ylab='TEMP',col='red',lwd=2)
text(Dataset$LATITUDE,Dataset$TEMP+1,row.names(Dataset)) # add the names of the points
```

Categorial variable

- Contingency table for one categorial variable

In [ ]:
```
tab=table(Dataset$GEO)
tab
pie(tab)
```

- Contingency table for two categorial variables

In [ ]:
```
shiny=Dataset[,31]>2000&Dataset[,1]>50 # Creation of a new categorial variable
shiny
shiny=as.factor(shiny)
tab=table(shiny,Dataset$GEO)
tab
barplot(tab)
```

PCA

FactoMineR is a package dedicated to multivariate analysis. We need to load it in the R session :

In [ ]:
```
library(FactoMineR)
```

The function *PCA* creates a dataframe with the results of the principal analysis components

```
PCA(Dataset, scale.unit=TRUE, ncp= ...)
```

For more details about PCA type help(PCA).

First, we create a subset of the dataset with the variables used in the PCA.

In [ ]:
```
Dataset.PCA=Dataset[, c('NO2' ,'DENSITY' ,'JANr' ,'FEBr' ,'MARr' ,'APRr' ,'MAYr' ,'JUNr' ,'JULr' ,'AUGr', '
SEPr', 'OCTr' ,'NOVr' ,'DECr' ,'RAINFALL' ,'JANdr' ,'FEVdr' ,'MARdr' ,'APRdr' ,'MAYdr', 'JUNdr' ,'JULdr' ,'
AUGdr', 'SEPdr', 'OCTdr', 'NOVdr' ,'DECdr', 'DAYS_RAINFALL', 'TEMP', 'TEMP_RANGE' ,'SUNSHINE', 'LATITUDE' ,
'LONGITUDE')] # c(a,b,c) creates the vector

res=PCA(Dataset.PCA,scale.unit=TRUE,ncp=5,graph=F)
attributes(res)
```

- **Eigenvalues**

res$eig gives a matrix containing all the eigenvalues of the covariance matrix, the percentage of variance and the cumulative percentage of variance

Here, the three first components contain more than 87% of the information. We can reduce the space to dimension 3.

In [ ]:
```
# display the eigenvalues
res$eig
# barplot of the percentages of variance explained by the principal axes
barplot(res$eig[,2])
```

- **Variables**

res$var gives a list of matrices containing all the results for the active variables (coordinates, correlation between variables and axes, square cosine, contributions).

We can see that Axis 1 is mainly built with variables related to the number of wet days (annual and monthly) variables, and Axis 2 is built with rainfall variables.

In [ ]:
```
# display the results for variables
res$var
```

plot.PCA with argument *choix='var'* plot the variables in the correlation circle for the axes selected with argument *axis=c(.,.)*.

In [ ]:
```
# plot the correlation circle for axes 1 and 2
plot.PCA(res,choix="var",axes=c(1,2))
# plot the correlation circle for axes 1 and 4
plot.PCA(res,choix="var",axes=c(1,4))
```

We can observe :

. a negative correlation between DAYS_RAINFALL and TEMP and SUNSHINE. When the number of rainy days increases the number of shiny days and the temperature decrease !!!

. Variables NO2,TEMP_RANGE, LONGITUDE and DENSITY are not represented well by the two first axes. If we want a good representation of NO2, we need Axis 4.

. two clusters of variables, one with the variables related to the number of wet days and the other with the variables related to the rainfall.

- **Supplementary variables**

It is possible to add supplementary variables. They don't contribute to the computation of the principal axes but they are represented in the correlation circle.

Here we remove "MONTH"+r and "MONTH"+dr variables from the computation of the PCA and we add them as supplementary variables. It changes the principal axes.

In [ ]:
```
# Add variables
res1=PCA(Dataset.PCA , scale.unit=TRUE, ncp=5, quanti.sup=c(3:14,16:27), graph   = FALSE)
plot.PCA(res1, axes=c(1, 2), choix='var', col.var='black', col.quanti.sup='red')
```

- **Individuals**

res$ind gives a list of matrices containing all the results for the active individuals (coordinates, square cosine, contributions)

In [ ]:
```
# display the results for individuals
```

plot.PCA with argument *choix='ind'* plot the principal components

In [ ]:
```
# plot the principal components on axes 1 and 2
plot.PCA(res,choix="ind",axes=c(1,2))
plot.PCA(res,choix="var",axes=c(1,2))
```

- **Outliers and supplementary individuals**

Biarritz (near Bayonne on the map) is far from the gravity center and may be an outlier. We remove it from the dataset for the computation of the principal axes. But we add it as supplementary point.

In [ ]:
```
res2<-PCA(Dataset.PCA, scale.unit=TRUE, ncp=5, ind.sup=5,graph = FALSE)
plot.PCA(res2, axes=c(1, 2), choix='ind', col.ind.sup='blue', label=c('ind','ind.sup'))
plot.PCA(res2,axes=c(1,2),choix='var',title='Without Biarritz')
plot.PCA(res,axes=c(1,2),choix='var',title='With Biarritz')
```

In [ ]:
```
plot.PCA(res,choix="ind",axes=c(1,4))
plot.PCA(res,choix="var",axes=c(1,4))
```

Remove Biarritz from the dataset doesn't change the results. It means that Biarritz is not an outlier but just a city with a high level of rainfall like Brest.

Whereas Marseille, Toulon and Nice are shiny and hot cities.

If we plot the results for Axe4, we see that Paris is a polluted city.

We can say nothing about the cities near the center of the graph (=the projection of the gravity center). Maybe they are near the gravity center, and maybe the are very far but orthogonal to the 2D graph.

In order to understand the impact of an outlier, we use PCA with the dataset with the outlier error (Dataset0).

In [ ]:
```
Dataset0=subset(Dataset,Dataset$TEMP>0)
Dataset0.PCA=Dataset0[, c('NO2' ,'DENSITY' ,'JANr' ,'FEBr' ,'MARr' ,'APRr' ,'MAYr' ,'JUNr' ,'JULr' ,'AUGr',
 'SEPr', 'OCTr' ,'NOVr' ,'DECr' ,'RAINFALL' ,'JANdr' ,'FEVdr' ,'MARdr' ,'APRdr' ,'MAYdr', 'JUNdr' ,'JULdr'
,'AUGdr', 'SEPdr', 'OCTdr', 'NOVdr' ,'DECdr', 'DAYS_RAINFALL', 'TEMP', 'TEMP_RANGE' ,'SUNSHINE', 'LATITUDE'
 ,'LONGITUDE')] # c(a,b,c) creates the vector
res0=PCA(Dataset0.PCA,scale.unit=T,ncp=5,graph=F)
plot.PCA(res0,axes=c(1,2),choix='var',title='With ERROR')
```

- **Supplementary factors**

We add supplementary individuals *South*, *West*, *North* and *East* with the factor GEO.

In [ ]:
```
# Add factors
res<-PCA(Dataset , scale.unit=TRUE, ncp=5, quali.sup=c(34: 34), graph =  FALSE)
plot.PCA(res, axes=c(1, 2), choix='ind', col.quali='red',label=c('quali','ind'))
```

Clustering

- *scale* reduced and centered the variables
- *dist* creates the distance matrix
- *hclust(distance matrix)* hierarchical clustering
- *kmeans(dataset)* k-means clustering

In [ ]:
```
#Clustering with the 33 quantitative variables
Dataset1.CLUST=scale(Dataset[,1:33],center=T,scale=T)
# Matrix of distances
#help(dist)
distance1=dist(Dataset1.CLUST)
# CAH
cah.res1=hclust(distance1,method="ward.D2")
# plot the criterion of dissimilarity
plot(cah.res1$height)
```

In [ ]:
```
# plot the dendrogram
plot(cah.res1)
nclust=4 # number of clusters
# Add the clusters to the dendrogram
rect.hclust(cah.res1,k=nclust)
# create a vector with the clusters
cluster.cah =cutree(cah.res1,k=nclust)
```

In [ ]:
```
# Clustering with the fisrt principal axes
Dataset2.CLUST=scale(res$ind$coord[,1:4],center=T,scale=T)
# Matrix of distances
distance2=dist(Dataset2.CLUST)
# CAH
cah.res2=hclust(distance2,method="ward.D2")
plot(cah.res2$height)
plot(cah.res2)
# The number of clusters depends on the number of principal axes. We can see the difference if we take the
two fisrt principal axes or the for fisth.
```

In [ ]:
```
nclust=5 # number of clusters
# create a vector with the clusters
cluster.cah =cutree(cah.res2,k=nclust)

# define a new categorial variable in the data
component=as.data.frame(res$ind$coord)
component$CLUSTER=as.factor(cluster.cah)
cluster1=subset(component,CLUSTER==1)
cluster2=subset(component,CLUSTER==2)
cluster3=subset(component,CLUSTER==3)
cluster4=subset(component,CLUSTER==4)
cluster5=subset(component,CLUSTER==5)

range.graph=apply(res$ind$coord,2,range)
axis1=1
axis2=3
plot(cluster1[,axis1],cluster1[,axis2],xlim=range.graph[,axis1],ylim=range.graph[,axis2],pch='*',col='red')
text(cluster1[,axis1],cluster1[,axis2],row.names(cluster1),col='red') # add the labels
points(cluster2[,axis1],cluster2[,axis2],pch='*',col='blue') # add points to an existing Figure
text(cluster2[,axis1],cluster2[,axis2],row.names(cluster2),col='blue')
points(cluster3[,axis1],cluster3[,axis2],pch='*',col='green')
text(cluster3[,axis1],cluster3[,axis2],row.names(cluster3),col='green')
points(cluster4[,axis1],cluster4[,axis2],pch='*',col='brown')
text(cluster4[,axis1],cluster4[,axis2],row.names(cluster4),col='brown')
points(cluster5[,axis1],cluster5[,axis2],pch='*',col='black')
text(cluster5[,axis1],cluster5[,axis2],row.names(cluster5),col='black')
```